

Connecting Industrial NL Applications to Knowledge (in Nuance)

Peter F, Patel-Schneider

NAIL Laboratory, Nuance Communications, Sunnyvale, CA, USA

Extended Abstract

It seems natural to have speech- and natural language-enabled applications utilize knowledge repositories. The usual situation for fielded industrial applications is quite different, however, with applications employing data sources that are specific to particular, limited tasks.

Fielded speech- and natural language-enabled industrial applications are generally purpose-built to solve a limited task, such as order processing or answering user questions about a product or service. Even more-general fielded systems are built from combinations of these limited applications, with little or no communication between these component applications. The data that the application uses is limited to the task at hand. There is no perceived need for the application to access a wide-coverage knowledge repository and using information from such a repository would require major changes to the application.

The interface between the speech and natural language processing part of fielded speech- and natural language-enabled industrial applications generally takes the form of a small set of commands, each with a small number of arguments. Each user input is transformed into one of the commands plus text values for each of its arguments. The command is then directly processed by the application (or by one of a set of available actions). The application is responsible for interpreting the text values for the command arguments. There is no place in this interface for complex user inputs whose processing might benefit from access to a knowledge repository.

If there is an exception to these rules it takes the form of a general-purpose but simple question answering component. This component may access general knowledge but not in a way that is useful to the rest of the system.

In my group in Nuance we are producing a framework for applications that eliminate these impediments to the use of knowledge repositories. The framework uses a speech and natural language component that produces a logical form for the user input. Central to the framework is a dialog manager that can take this richer interface and build responses from a set of primitive actions. One of these primitive actions can be to look for information in a wide-coverage knowledge repository. The dialog manager can take these results and pass them to other components of the system. This gives purpose-built applications the ability to utilize a knowledge repository, for example to select entities in its area of expertise by out-of-domain features of the entity.